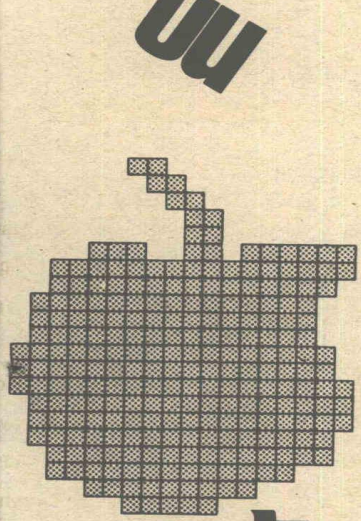
กราฟิกภาพหยาบ



แอปเปิ้ลปาสคาล

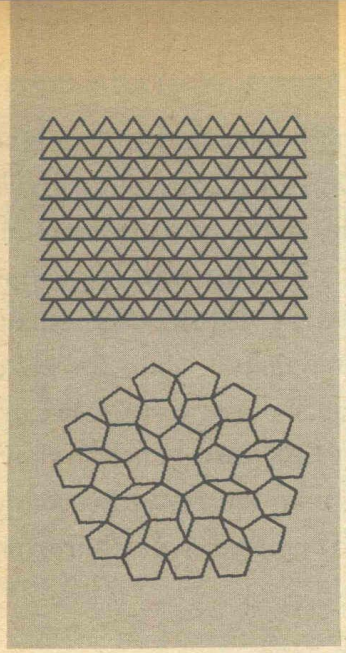
อำพล สงวนศิริธรรม

ตามปกติการแสดงภาพกราฟิกในแอปเปิ้ลปาสคาล จะแสดงได้แต่เพียงกราฟิกภาพละเอียด(High-resolution Graphics) เท่านั้น โดยอาศัย TURTLEGRAPHICS UNIT ที่มีอยู่แล้วในไฟล์ชื่อ SYSTEM.LIBRARY ในแผ่นโปรแกรมแอปเปิ้ลปาสคาลนั่นเอง ถ้าต้องการจะให้แสดงกราฟิกภาพหยาบ(Low-resolution Graphics) ได้ด้วยจำเป็นต้องสร้างโปรแกรมที่จะแสดงคุณสมบัติด้านนี้ขึ้นใช้เอง

ผู้เขียนขอเสนอโปรแกรมที่ช่วยทำให้แอปเปิ้ลปาสคาลสามารถแสดงกราฟิกภาพหยาบได้ โดยการสร้างโปรแกรมที่มีลักษณะ UNIT รวม 2 โปรแกรม โปรแกรมแรกคือโปรแกรมชื่อ PEEKPOKE โปรแกรมนี้ประกอบด้วย procedure ชื่อ poke และ function ชื่อ peek จะทำหน้าที่เช่นเดียวกับคำสั่ง PEEK และ POKE ในภาษาเบสิก โปรแกรมที่สองชื่อ LOWRESGR โปรแกรมนี้จะสร้าง procedures ที่จำเป็นสำหรับกราฟิก

1

```
(*$s+*)
UNIT peekpoke;

  INTERFACE

TYPE

  byte          =       0..255;
  memloc        = PACKED  ARRAY[0..1] OF byte;

  access        = RECORD CASE BOOLEAN OF

                    false:  (address: INTEGER);
                    true:   (pointer: ^memloc);

                  END;

VAR

  memory:       access;

PROCEDURE poke(spot: INTEGER; value: byte);
FUNCTION  peek(spot: INTEGER): byte;

  IMPLEMENTATION

PROCEDURE poke;
BEGIN

  WITH memory DO
  BEGIN

    address     := spot;
    pointer^[0] := value;

  END  (* WITH memory DO *);

END  (* PROCEDURE poke *);

FUNCTION peek;
BEGIN

  WITH memory DO
  BEGIN
    address     := spot;
    peek        := pointer^[0];

  END  (* WITH memory DO *);

END  (* FUNCTION peek *);

BEGIN
END  (* UNIT peekpoke *).
```



ภาพหยาบ คือ introwaddr, setcolor, plot, hline, vline, settext และ gr โดยเฉพาะใน procedure gr จะทำหน้าที่เหมือนชุดคำสั่งในภาษาเบสิกคือ

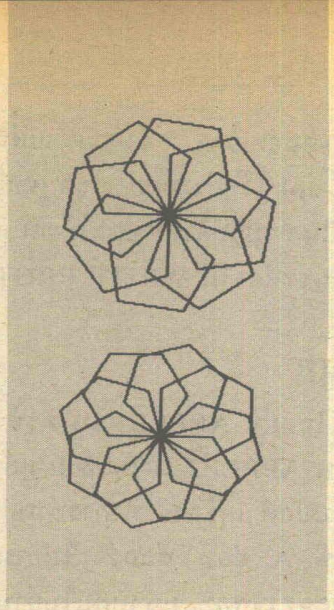
```
GR
POKE -16302,0
COLOR = 0
FOR I = 40 TO 47
HLIN 0,39 AT I
NEXT I
```

ซึ่งหมายถึงการเข้าสู่โหมดกราฟิกภาพหยาบชนิดเต็มจอนั่นเอง นอกจาก procedure ต่าง ๆ แล้ว ยังประกอบด้วย function ชื่อ screen รายละเอียดต่าง ๆ ขอให้ดูจากโปรแกรมหมายเลข 1 ในรูปที่ 1 และโปรแกรมหมายเลข 2 ในรูปที่ 2 สำหรับความหมายของคำต่าง ๆ ในโปรแกรมขอให้ศึกษาจากคู่มือแอปเปิ้ลปาสคาลและบทความเรื่อง "ปฐมบทของปาสคาล" โดยอาจารย์บุญเลิศ เอี่ยมทัศนา ในวารสาร"ไมโครฯ" ตั้งแต่ฉบับที่ 10 เป็นต้นมา)

และเพื่อเป็นตัวอย่างการใช้ประโยชน์จากสองโปรแกรมแรก ขอเสนอโปรแกรมหมายเลข 3 ในรูปที่ 3 โปรแกรมนี้จะใช้ UNIT ทั้งสองจากสองโปรแกรมแรก และสร้างภาพเป็นตัวอักษรคำว่า "สวัสดี" ด้วยสีชมพู (ถ้าเป็นมอนิเตอร์สี) ที่กลางจอภาพ

2

```
(*$s+*)
UNIT lowresgr;

  INTERFACE

USES (*$u APPLE1:peekpoke.code*) peekpoke;

CONST

  lores          =        -16298;
  graphics       =        -16304;
  mixtext        =        -16301;
  alltext        =        -16303;
  fullscreen     =        -16302;

  black          =             0;
  magenta        =             1;
  darkblue       =             2;
  purple         =             3;
  darkgreen      =             4;
  grey           =             5;
  medblue        =             6;
  lightblue      =             7;
  brown          =             8;
  orange         =             9;
  alsogrey       =            10;
  pink           =            11;
  green          =            12;
  yellow         =            13;
  aqua           =            14;
  white          =            15;

TYPE

  colorval       =         0..15;

VAR

  color:                colorval;

  rowaddress:    ARRAY[0..23] OF INTEGER;

PROCEDURE initrowaddr;
PROCEDURE setcolor(c:colorval);
PROCEDURE plot(row: INTEGER; col: INTEGER);
PROCEDURE hline(col1,col2: INTEGER; row: INTEGER);
PROCEDURE vline(row1,row2: INTEGER; col: INTEGER);
PROCEDURE settext;
PROCEDURE gr;    (* note: set full screen *)
FUNCTION  screen(row: INTEGER; col: INTEGER): colorval;
  IMPLEMENTATION

PROCEDURE initrowaddr;
VAR
  i: INTEGER;
BEGIN
  rowaddress[0] := 1024;
  rowaddress[1] := 1152;
  rowaddress[2] := 1280;
  rowaddress[3] := 1408;
  rowaddress[4] := 1536;
  rowaddress[5] := 1664;
  rowaddress[6] := 1792;
  rowaddress[7] := 1920;
  rowaddress[8] := 1064;
  rowaddress[9] := 1192;
  rowaddress[10] := 1320;
  rowaddress[11] := 1448;
  rowaddress[12] := 1576;
```



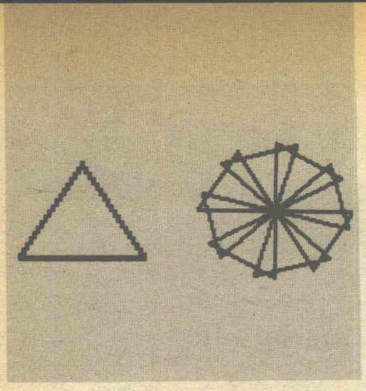
```
rowaddress[13] := 1704;
rowaddress[14] := 1832;
rowaddress[15] := 1960;
rowaddress[16] := 1104;
rowaddress[17] := 1232;
rowaddress[18] := 1360;
rowaddress[19] := 1488;
rowaddress[20] := 1616;
rowaddress[21] := 1744;
rowaddress[22] := 1872;
rowaddress[23] := 2000;

ND  (* PROCEDURE initrowaddr *);

ROCEDURE setcolor;
EGIN

 color := c;

ND  (* PROCEDURE setcolor *);

ROCEDURE plot;
AR
 hi,lo: 0..15;
 old: byte;
 spot: INTEGER;
EGIN

 spot   := rowaddress[row DIV 2] + col;

 old    := peek(spot);
 hi     := old DIV 16;
 lo     := old MOD 16;

 IF (row MOD 2) = 0
 THEN
   poke(spot,color+16*hi)
 ELSE
   poke(spot,lo+color*16)
 (* endif *);

ND (* PROCEDURE plot *);

ROCEDURE hline;
AR
 i: INTEGER;
```

►

2

```
BEGIN

  FOR i := col1 TO col2 DO
    plot(row,i)
  (* enddo *);

END  (* PROCEDURE hline *);
PROCEDURE vline;
VAR
  i: INTEGER;
BEGIN

  FOR i := row1 TO row2 DO
    plot(i,col)
  (* enddo *);

END  (* PROCEDURE vline *);

PROCEDURE settext;
BEGIN

  poke(alltext,0);

END (* PROCEDURE settext *);

PROCEDURE gr;
VAR
  i: INTEGER;
BEGIN

  poke(lores,0);
  poke(fullscreen,0);
  poke(graphics,0);
  WITH memory DO
  BEGIN

    address := 1024;
    fillchar(pointer^[0],1024,chr(0));

  END  (* WITH memory DO *);

END  (* PROCEDURE gr *);

FUNCTION screen;
VAR
  hi,lo: 0..15;
  old: byte;
  spot: INTEGER;
BEGIN

  spot := rowaddress[row DIV 2] + col;
  old  := peek(spot);
  hi   := old DIV 16;
  lo   := old MOD 16;


  IF (row MOD 2) = 0
  THEN
    screen := hi
    ELSE
      screen := lo
    (* endif *);

END  (* FUNCTION screen *);

BEGIN

  initrowaddr;

END.
```

■



```
PROGRAM swasdee;                                    3

USES (*$u APPLE1:PEEKPOKE.CODE*)
                              peekpoke,
     (*$u APPLE1:LOWRES.CODE*)
                              lowresgr;
VAR


  ch:              CHAR;

BEGIN



  gr;

  setcolor(pink);
  hline(7,10,16);
  plot(14,11);
  plot(15,11);
  vline(17,22,11);
  hline(8,10,18);
  vline(19,22,7);
  vline(21,22,8);

  setcolor(pink);
  vline(21,22,17);
  vline(17,22,18);
  hline(15,17,16);
  plot(17,14);

  setcolor(pink);
  plot(11,15);
  plot(12,14);
  hline(15,16,13);
  plot(12,17);
  plot(11,18);

  setcolor(pink);
  vline(21,22,22);
  vline(19,22,21);
  hline(22,24,18);
  vline(17,22,25);
  hline(21,24,16);
  plot(14,25);
  plot(15,25);

  setcolor(pink);
  hline(29,30,20);
  plot(21,30);
  plot(22,29);
  vline(17,22,28);
  hline(29,31,16);
  vline(17,22,32);

  setcolor(pink);
  hline(28,32,13);
  plot(12,28);
  hline(29,31,11);
  vline(10,13,32);

  readln(ch);
  settext;
  write(chr(12));
  EXIT(PROGRAM);

END.
```

คำแนะนำสำหรับผู้เริ่มต้น

ผู้ที่จะใช้โปรแกรมดังกล่าวนี้ได้ จะต้องมีอุปกรณ์ดังต่อไปนี้คือ

1. ไมโครคอมพิวเตอร์แอปเปิ้ลทู หรือเครื่องเลียนแบบ มีหน่วยความจำบนเมนบอร์ด 48K RAM และมีการ์ด 16 K RAM เสียบอยู่ที่สล้อต #0
2. ควรใช้ดิสก์ไดรฟ์ 2 เครื่อง
3. แผ่นโปรแกรมแอปเปิ้ลปาสคาล จำนวน 3 แผ่น คือ APPLE1, APPLE2, APPLE2 (ถ้ามีแผ่น APPLE 0 ด้วยก็ดี)
4. คู่มือการใช้แอปเปิ้ลปาสคาล 2 เล่ม ได้แก่ Apple Pascal : Language Reference Manual และ Apple Pascal : Operating System Reference Manual 1
5. ถ้ามี 80 คอลัมน์อยู่ด้วยจะทำให้แสดงผลบน 80 คอลัมน์โดยอัตโนมัติ ซึ่งสะดวกในการพิมพ์คำสั่งยาว ๆ แต่ถ้าไม่มีก็ใช้ได้ กรณีนี้ต้องใช้ CTRL-A เพื่อดูอีก 40 คอลัมน์ที่เหลือ

คำเตือนก่อนเริ่มต้น

เนื่องจากขณะใช้งาน แผ่นโปรแกรมแต่ละแผ่นจะมีการอ่านและการเขียน เป็นระยะๆ ตลอดเวลา จึงควรทำแผ่นสำร
(back-up) ไว้อย่างน้อย 1 ชุด และใช้แ
สำรองในการปฏิบัติงาน ขณะใช้งาน และดิ
ทุกแผ่นจะต้องไม่ WRITE PROTEC

การบูท

ใส่แผ่น APPLE1 ในไดรฟ์ A แ
ใส่แผ่น APPLE2 ในไดรฟ์ B แล้วทำ
บูทตามปกติ เครื่องจะทำการอ่านแผ่นดิ
ที่ไดรฟ์ A ก่อน สักครู่จะไปอ่านที่ได
B เพื่อตรวจสอบ แล้วกลับไปอ่านที่ได
A อีกครู่หนึ่งบนจอภาพจะมีลักษณะดังรูป

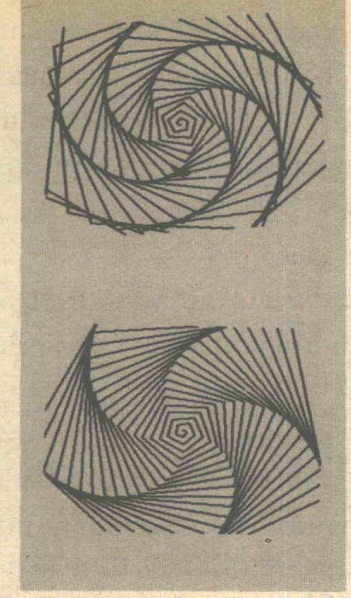
บรรทัดบนสุดเรียกว่า Prompt L
(ซึ่งถ้าในระบบคอมพิวเตอร์ของท่าน
80 คอลัมน์ จะเห็นข้อความเพียง 40 คอ
แรกคือ

COMMAND:E(DIT, R(UN, F(ILE, C(OMP,
ซึ่งถ้าต้องการดูข้อความที่เหลือ ใ
CTRL-A) หลังจากนั้น ถ้าพิมพ์เครื่องห
คำถาม (?) จะได้ข้อความที่เหลืออีกตั

COMMAND:U(SER RESTART, I(NITIAL
H(ALT

ข้อความใน Prompt Line ขณะนี้
คำสั่งใช้งานในระบบทั้งหมด เรียกว่า C
MAND LEVEL

```
                    4
Command: E(dit, R(un, F(ile, C(omp, L(ink,
         X(ecute, A(ssem, D(ebug, ?[1.1]



Welcome APPLE1, to APPLE II Pascal 1.1
Base on UCSD Pascal II.1
Current date is 18-Feb-86



(C) Apple Computer Inc. 1979, 1980
(C) U.C.Regents 1979
```



มต้นใช้งาน

ขั้นที่ 1 จาก COMMAND LEVEL E เพื่อเข้าสู่ EDITOR MODE สักครู่ ompt Line จะเปลี่ยนเป็นรูป A

```
A
EDIT:
 WORKFILE IS PRESENT.FILE (<RET>
R NO FILE<ESC-RET>TO EXIT)
```

ดคีย์รีเทอร์นจะเห็น Prompt Line เปลี่ยน รูป B

```
B
DIT:A(JUST C(PY D(LETE F(IND I(NSRT
P R(PLACE Q(UIT X(CHNG.
```

ก Prompt Line ขณะนี้ว่า EDITOR MMANDS

ขั้นที่ 2 จาก EDITOR COMMANDS I เพื่อเข้าสู่ Insert mode จะเห็น Prompt e เปลี่ยนเป็นรูป C

```
C
NSERT:TEXT[<BS>A CHAR,<DEL>A
E ][<ETX>ACCEPT,<ESC>...
```

Prompt Line ขณะนี้ว่า INSERT MMANDS

ขณะนี้พร้อมที่จะพิมพ์โปรแกรมหมาย 1 ได้แล้ว มีคำแนะนำเพิ่มเติมดังนี้

(1) พิมพ์ข้อความตามปกติ ถ้าพิมพ์ผิดใช้คีย์ < - ในการลบถอยหลังทีละตัว และใช้คีย์ - >└ในการเขียนตัวอักษรนั้นกลับมา

(2) พิมพ์คำสั่งในแต่ละบรรทัดจบให้กดคีย์รีเทอร์นจะขึ้นบรรทัดใหม่ และยังอยู่ใน INSERT MODE

(3) ถ้ากดคีย์รีเทอร์นไปแล้วพบว่าพิมพ์ผิดหรือตก ต้องการกลับไปแก้ไขให้ออกจาก INSERT MODE ก่อน และไปแก้ไขใน EDIT MODE

(4) ต้องการออกจาก INSERT MODE ให้กด CTRL-C (กดคีย์ CONTROL และ คีย์ C พร้อมกัน) จะออกไปสู่ EDIT MODE

การแก้ไขขณะอยู่ใน EDIT MODE

ถ้าต้องการเปลี่ยนแปลงแก้ไขข้อความที่พิมพ์ไปแล้ว ขณะที่อยู่ใน EDIT MODE ให้นำเคอร์เซอร์เลื่อนไปทับตัวอักษรแรกของคำหรือข้อความก่อน การเลื่อนเคอร์เซอร์จะใช้คีย์ดังนี้

(1) โดยการกดคีย์รีเทอร์นจะนำเคอร์เซอร์ไปไว้ยังอักษรตัวแรกของบรรทัดถัดลงไป 1 บรรทัด

(2) ใช้คีย์ < - หรือ - > ในการเลื่อนเคอร์เซอร์ไปทางซ้าย หรือทางขวาตามปกติ

(3) ใช้ CTRL-L เพื่อเลื่อนเคอร์เซอร์ลงในแนวดิ่ง

(4) ใช้ CTRL-O เพื่อเลื่อนเคอร์เซอร์ขึ้นในแนวดิ่ง

เมื่อเคอร์เซอร์อยู่ ณ ตำแหน่งที่ต้องการแล้ว มีทางเลือก 2 ทางคือ

(1) กดคีย์ 1 จะเข้าสู่ INSERT MODE และถ้าต้องการยุติให้กด CTRL-C หรือ

(2) กดคีย์ D จะเข้าสู่ DELETE MODE เพื่อลบข้อความที่ไม่ต้องการออก เสร็จแล้วให้กด CTRL-C

การกด CTRL-C ทั้งสองกรณี จะกลับคืนไปที่ EDIT MODE

ถ้าต้องการออกจาก EDIT MODE ให้กด Q (คือ QUIT) บนจอภาพจะมีข้อความดังรูปที่ 5

```
5
> QUIT :
  U(PDATE THE WORKFILE AND LEAVE
  E(XIT WITHOUT UPDATING
  R(RETURN TO THE EDITOR WITHOUT
  UPDATING
  W(RITE TO A FILE NAME AND RETURN
  S(AVE WITH SAME NAME AND RETURN
```

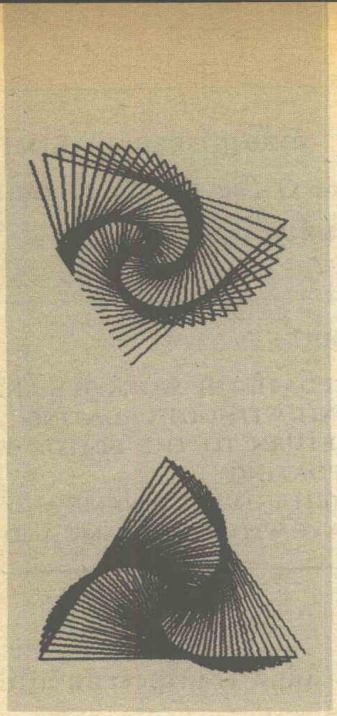
ให้กด U เครื่องจะบันทึกโปรแกรมที่มีอยู่ในหน่วยความจำขณะนั้น บนแผ่น APPLE1 ในไดรฟ์ A โดยมีชื่อไฟล์นี้ว่า SYSTEM.WRK.TEXT โดยอัตโนมัติ แล้วจะกลับไปที่ COMMAND LEVEL และต่อจากนี้ถ้ากด E เพื่อเข้าสู่ EDIT MODE เครื่องจะอ่านโปรแกรมชื่อ SYSTEM.WRK.TEXT ให้ทุกครั้ง จะทำให้การแก้ไขเปลี่ยนแปลงโปรแกรมทำได้โดยสะดวก

สมมติว่าจนถึงขณะนี้ท่านผู้อ่านได้คีย์โปรแกรมหมายเลข 1 และเก็บบันทึกไว้ในไฟล์ชื่อ SYSTEM.WRK.TEXT เรียบร้อยแล้ว

การคอมพายล์โปรแกรมหมายเลข 1

ในการคอมพายล์จะต้องใช้ไฟล์ที่ชื่อว่า SYSTEM.COMPILER ซึ่งอยู่ในแผ่น APPLE2 (ซึ่งขณะนี้อยู่ในไดรฟ์ B เรียบร้อยแล้ว) จาก COMMAND LEVEL ให้กด C จะเริ่มต้นการคอมพายล์ทันที และระหว่างการคอมพายล์ถ้ามีการผิดพลาดจะมีข้อความบอกถึงสาเหตุของความผิดพลาด และให้โอกาสในการกลับเข้าสู่ EDIT MODE เพื่อไปทำการแก้ไขได้ทันที เมื่อแก้ไขเสร็จแล้วให้กลับมาเริ่มต้นคอมพายล์ใหม่

ถ้าคอมพายล์เรียบร้อยแล้ว บรรทัดสุดท้ายจะบอกจำนวนไบท์ที่ใช้ทั้งหมด และขณะนี้บนแผ่น APPLE1 จะได้ไฟล์ที่ผ่านการคอมพายล์สมบูรณ์แล้วชื่อ SYSTEM.WRK.CODE



ก่อนพิมพ์โปรแกรมหมายเลข 2 ต้องเปลี่ยนชื่อไฟล์ที่ชื่อ SYSTEM.WRK.TEXT และ SYSTEM.WRK.CODE ก่อนดังนี้

1) จาก COMMAND LEVEL กด F เพื่อเข้าสู่ FILER COMMANDS จะเห็น Prompt Line เป็นดังรูป D

FILER: G(ET, S(AVE, W(HAT, N(EW, L(DIR, R(EM, C(HNG, T(RANS, D(ATE,

2) ให้กด C จะมีข้อความว่า

CHANGE?

ให้พิมพ์ APPLE1:SYSTEM.WRK.TEXT แล้วกดรีเทอร์นจะมีข้อความ

CHANGE TO WHAT?

ให้พิมพ์ APPLE1:PEEKPOKE.TEXT ตามด้วยกดรีเทอร์นจะเห็นเริ่มทำงาน เมื่อเปลี่ยนชื่อไฟล์เสร็จ จะมีข้อความแจ้งให้ทราบ และกลับไปที่ FILER COMMANDS

ให้ใช้วิธีเดียวกันนี้เปลี่ยนชื่อไฟล์ที่ชื่อ SYSTEM.WRK.CODE ให้เป็น PEEKPOKE.CODE (ซึ่งอยู่บนแผ่น APPLE1 ตามเดิม)

เหตุที่ต้องเปลี่ยนชื่อไฟล์ในตอนนี้ ก็เพราะว่าทุกครั้งที่เริ่มพิมพ์โปรแกรมใหม่และ SAVE ไว้บนแผ่นปาสคาลจะใช้ชื่อไฟล์ว่า SYSTEM.WRK.TEXT เสมอ ซึ่งถ้าไม่เปลี่ยนชื่อไฟล์เดิมก่อน การบันทึกโปรแกรมใหม่จะไปทับโปรแกรมเดิม

เริ่มพิมพ์โปรแกรมหมายเลข 2

การพิมพ์โปรแกรมหมายเลข 2 ให้ใช้วิธีเดียวกับโปรแกรมหมายเลข 1 เมื่อพิมพ์เสร็จแล้วให้ทำการคอมพายล์ มีข้อควรระวังคือ โปรแกรมหมายเลข 2 ได้กำหนดให้ใช้ UNIT ชื่อ PEEKPOKE.CODE ซึ่งต้องอยู่บนแผ่น APPLE1 ขณะคอมพายล์จึงจะสามารถคอมพายล์ได้

เมื่อคอมพายล์เสร็จเรียบร้อยแล้ว ให้เปลี่ยนชื่อไฟล์ที่ชื่อ SYSTEM.WRK.TEXT เป็น LOWRES.TEXT และเปลี่ยนชื่อไฟล์จาก SYSTEM.WRK.CODE เป็น LOWRES.CODE

โปรแกรมหมายเลข 3

เป็นโปรแกรมตัวอย่างในการใช้งาน เมื่อพิมพ์โปรแกรมนี้เสร็จแล้ว ให้ทำการคอมพายล์ตามปกติ โดยขณะทำการคอมพายล์จะต้องมีไฟล์ชื่อ PEEKPOKE.CODE และ LOWRES.CODE อยู่บนแผ่น APPLE1 จึงจะคอมพายล์ได้

เนื่องจากโปรแกรมหมายเลข 3 เป็นโปรแกรมที่ใช้ UNIT ดังนั้นจะไม่สามารถ EXECUTE ได้ทันทีจะต้องทำการลิงค์กับโปรแกรม UNIT นั้นก่อน ซึ่งมีวิธีการดังนี้

การลิงค์

ในการลิงค์ต้องใช้ไฟล์ที่ชื่อ SYSTEM.LINKER ซึ่งอยู่ในแผ่น APPLE2 (ซึ่งในขณะนี้อยู่ในไดรฟ์ B เรียบร้อยแล้ว)

จาก COMMAND LEVEL ให้กด L จะมีข้อความ

HOST?

ให้พิมพ์ APPLE1:SYSTEM.WRK.CODE(คือโปรแกรมหมายเลข 3 ที่ผ่านการคอมพายล์แล้ว นั่นเอง) พิมพ์เสร็จกดรีเทอร์น จะมีข้อความ

LIB?

ให้พิมพ์ APPLE1:PEEKPOKE.CODE(เป็นชื่อ UNIT แรก ที่ระบุในโปรแกรมหมายเลข 3) พิมพ์เสร็จกดรีเทอร์นจะมีข้อความ

LIB?

ให้พิมพ์ APPLE1:LOWRES.CO
(เป็นชื่อ UNIT ที่สอง ที่ระบุในโปรแก
หมายเลข 3) พิมพ์เสร็จกดรีเทอร์นจะมี
ความ

LIB?

เนื่องจากได้พิมพ์ชื่อ UNIT ครบแ
ถึงตอนนี้ให้กดรีเทอร์นอย่างเดียวจะมีข้อค
ว่า

MAP FILE?

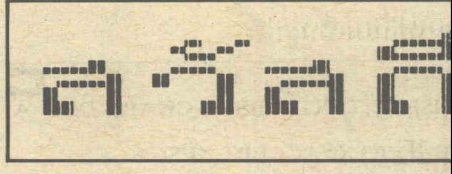
ถ้าต้องการข้อมูลเพื่อตรวจสอบการ
ในภายหลัง ให้พิมพ์ชื่อไฟล์ที่ต้องการล
ปาสคาลจะเปิดไฟล์ที่มีชื่อนี้ตามด้วย .TE
ซึ่งทำให้สามารถอ่านได้จาก EDIT MO
ถ้าไม่ต้องการ MAP ไฟล์นี้ ให้กดรีเทอ
อย่างเดียว จะมีข้อความว่า

OUTPUT?

ซึ่งหมายถึงให้ตั้งชื่อไฟล์ใหม่ ให้ที่
APPLE1:SAWASDEE(ซึ่งจะมีผลทำ
ไฟล์ที่ชื่อ SYSTEM.WRK.CODE บน
APPLE1 เปลี่ยนเป็น SAWASDE
CODE)เสร็จแล้วกดคีย์รีเทอร์น จะ
การ SAVE บนแผ่น APPLE1 หลังจาก
จะกลับไปยัง COMMAND LEVEL

การใช้งานโปรแกรมหลัก

จาก COMMAND LEVEL ให
X (หมายถึง EXECUTE) จะถามชื่อ
ให้พิมพ์ APPLE: SAWASDEE สั
ท่านจะเห็นคำว่า "สวัสดี" สีชมพู
เป็นมอนิเตอร์สี) อยู่กลางจอภาพ ตามรู
6



หวังว่าท่านผู้อ่านที่เริ่มต้นสนใจภ
ปาสคาล คงจะได้ประโยชน์และประสบก
ในบทความนี้ตามสมควร.